

CLAIMS

What is claimed is:

1. A method for processing timer events, the method comprising:
 - 5 receiving a timer subscription containing a time value and an identity of a module to notify upon expiration of the time value;
establishing a timer to track expiration of the time value;
detecting expiration of the timer;
in response to detecting expiration of the timer, determining if the module is
10 disabled, and if the module is disabled, enabling the module; and
notifying a subscriber in the module of expiration of the timer.
2. The method of claim 1 wherein the module includes a timer handler in the subscriber, the timer subscription further indicative of the timer handler, and notifying the
15 subscriber of the expiration of the timer further comprises invoking the indicated timer handler for execution.
3. The method of claim 1 wherein establishing the timer further comprises:
 - adding the identity of the module to a global timer map, the global timer map
20 operable to indicate a plurality of modules; and
adding a reference to the subscriber including the timer handler into a local timer map associated with the module.
4. The method of claim 3 wherein invoking further comprises:
 - 25 indexing, via the local timer map, a dispatch command operable to dispatch the timer handler.
5. The method of claim 4 wherein the local timer map includes an entry indicative of the subscriber including the timer handler within a module and the global timer map
30 includes an entry indicative of the module.

6. The method of claim 4 wherein the reference is a dynamic offset from a base to the location in a particular instantiation of the module, the base operable to change upon reenabling of the module.

5 7. The method of claim 2 wherein the expiration of the timer and resulting timer initiated invocation of the timer handler is independent of the enablement of the subscriber including the timer handler.

8. The method of claim 4 wherein determining if the module is disabled further
10 comprises:
employing the global timer map to find the entry corresponding to the timer expiration to determine the identity of the module corresponding to the timer event; and determining, from the identity of the module, if the module is disabled.

15 9. The method of claim 1 wherein the timer subscription is operable to indicate periodic and aperiodic expiration times.

10. The method of claim 1 wherein the subscription is received from a subscriber within the module, the subscriber including the timer handler.

20

11. The method of claim 1 wherein receiving the subscription further comprises receiving a subscription from multiple subscribers in the module, each subscriber operative to include a timer handler, further comprising, in response to detecting expiration of the timer, enabling disabled modules upon expiration of a timer subscribed
25 to by any of the multiple subscribers.

12. The method of claim 1 wherein subscription is a first subscription and includes a timer identity, further comprising receiving second subscription to the same timer as first subscription, the timer identified by a timer name provided by both the first subscription
30 and the second subscription.

13. The method of claim 1 further comprising resetting the expiration time value with an expiration time value from a second subscription for the same timer.

14. The method of claim 1 wherein

5 enabling modules corresponds to activation of a corresponding component by an activation mechanism; and

disabling corresponds to deactivation of the corresponding component by the activation mechanism, the activation and deactivation operations operable to reduce memory consumption by inactive components and provide selective invocation to
10 maintain availability of the component.

15. The method of claim 14 wherein enabling and disabling is performed at a level of granularity of the modules, each of the modules corresponding to a component and operable be enabled and disabled by activation and deactivation of the corresponding
15 component.

16. The method of claim 14 wherein activation and deactivation further comprises identifying, in a module server in communication with each of the modules, when to activate and deactivate modules based on information in the global timer map in the
20 component server.

17. The method of claim 14 wherein each of the modules is operable to include a plurality of threads, and disabling is performed by a thread manager operable to gracefully terminate each of the threads prior to deactivation, deactivation occurring by
25 informing each of the threads of the termination and computing when each thread has attained a termination point.

18. The method of claim 1 further comprising:
associating the timer with a generation counter, the generation counter
30 incrementally labeling each invocation from a particular subscriber;
comparing, upon completion of a timer handler, the generation counter;

canceling, if the generation counter indicates that the timer handler corresponds to the generation counter, the timer; and

maintaining, if the timer is periodic, the pending timer corresponding to the subscriber.

5

19. The method of claim 1 wherein associating the timer identity with a timer handler occurs in a native language of the timer handler and corresponding subscriber, and avoids a corresponding definition in an external interface language, the external interface language for generating timer specific code.

10

20. The method of claim 19 wherein the external interface language is the Object Management Group Interface Definition Language (OMG/IDL).

21. A method for time based invocation of subscribers comprising:

15 receiving a subscription to a timer, the timer associative of a timer handler and an expiration time, the timer handler having instructions operative for executing and completing a particular task and the expiration time indicative of performance of the time based task;

20 associating the timer with the timer handler, the association including a generic timer reference applicable to a plurality of timer handlers, the association operable to selectively enable a module including the timer handler upon expiration of the timer handler.

receiving an indication of expiration of the timer;
determining, via the association, the corresponding timer handler and the module
25 including the timer handler;
selectively enabling the module including the timer handler; and
dispatching the timer handler to execute and complete the time based task.

22. The method of claim 21 wherein, following selectively enabling:

30 enqueueing an indication of the timer expiration in a queue, the queue corresponding to the process including the module containing the subscriber; and

assigning, to a particular thread corresponding to the queue, performance of the timer handler corresponding to the expired timer.

23. The method of claim 21 further comprising
5 associating timer event data with each timer, and delivering the timer event data to the handler upon invocation of the timer handler; and
communicating, via the association of the timer handler and the timer, the timer data, the communicating of the data independent of the location of the timer handler.
- 10 24. An infrastructure server for processing timer events comprising:
a module server operable to receive a timer subscription containing a time value and an identity of a module to notify upon expiration of the time value;
a timer service in the module service operable to establishing a timer to track expiration of the time value, the timer service further operable to detecting expiration of
15 the timer, the timer service further operable to, in response to detecting expiration of the timer,
determining if the module is disabled, and if the module is disabled, enabling the module, and
notifying a subscriber in the module of expiration of the timer.
20
25. The infrastructure server of claim 24 wherein the module includes a timer handler in the subscriber, the timer subscription further indicative of the timer handler, and the timer service further operable to notify the subscriber of the expiration of the timer.
- 25 26. The infrastructure server of claim 25 further comprising a local timer map, the local timer service operable to invoke the subscriber by indexing, via the local timer map, a dispatch command operable to dispatch the timer handler.
27. The infrastructure server of claim 25 further comprising a global timer map,
30 wherein the timer service is operable to establish the timer by:

adding the identity of the module to the global timer map, the global timer map operable to indicate a plurality of modules; and

adding a reference to the subscriber including the timer handler into a local timer map associated with the module.

5

28. The infrastructure server of claim 27 wherein the local timer map includes an entry indicative of the subscriber including the timer handler within a module and the global timer map includes an entry indicative of the module.

10 29. The infrastructure server of claim 27 wherein the reference is a dynamic offset from a base to the location in a particular instantiation of the module, the base operable to change upon reenabling of the module.

15 30. The infrastructure server of claim 25 wherein the timer service is operable to invoke the subscriber and the corresponding timer handler is independently of the enabling of the subscriber including the timer handler.

20 31. The infrastructure server of claim 24 wherein the timer service is operable to receive a subscription from a subscriber within the module, the subscriber including the timer handler.

25 32. The infrastructure server of claim 24 wherein the module server is operable to enable modules by activating a corresponding component by an activation mechanism; and further operable to disable modules by deactivating the corresponding component by the activation mechanism, the activation and deactivation operations operable to reduce memory consumption by inactive components and provide selective invocation to maintain availability of the component.

30 33. The infrastructure server of claim 32 wherein enabling and disabling is performed at a level of granularity of the modules, each of the modules corresponding to a

component and operable be enabled and disabled by activation and deactivation of the corresponding component.

34. The infrastructure server of claim 32 wherein the module server is in
5 communication with each of the modules, and further operable to determine when to activate and deactivate modules based on information in the global timer map in module server.

35. The infrastructure server of claim 24 wherein each of the modules is operable to
10 include a plurality of threads, and disabling is performed by a thread manager operable to gracefully terminate each of the threads prior to deactivation, deactivation occurring by informing each of the threads of the termination and computing when each thread has attained a termination point.

15 36. The infrastructure server of claim 24 further comprising, for each timer, a generation counter, the timer service operable to associate each timer with the generation counter, the generation counter incrementally labeling each invocation from a particular subscriber, the timer service further operable to compare, upon completion of a timer handler, the generation counter; cancel, if the generation counter indicates that the timer
20 handler corresponds to the generation counter, the timer, and maintain, if the timer is periodic, the pending timer corresponding to the subscriber.

37. A computer program product having a computer readable medium operable to store computer program logic embodied in computer program code encoded thereon for
25 processing timer events in a services infrastructure, the method comprising:
computer program code for receiving a timer subscription containing a time value and an identity of a module to notify upon expiration of the time value;
computer program code for establishing a timer to track expiration of the time value;
30 computer program code for detecting expiration of the timer;

computer program code for, in response to detecting expiration of the timer,
determining if the module is disabled, and if the module is disabled, enabling the module;
and

5 computer program code for notifying a subscriber in the module of expiration of
the timer.

38. A computer data signal having program code for thereon for processing timer
events in a services infrastructure, the method comprising:

10 program code for receiving a timer subscription containing a time value and an
identity of a module to notify upon expiration of the time value;
program code for establishing a timer to track expiration of the time value;
program code for detecting expiration of the timer;
program code for, in response to detecting expiration of the timer, determining if
the module is disabled, and if the module is disabled, enabling the module; and
15 program code for notifying a subscriber in the module of expiration of the timer.

39. An infrastructure server for processing timer events comprising:

means for receiving a timer subscription containing a time value and an identity
of a module to notify upon expiration of the time value;
20 means for establishing a timer to track expiration of the time value;
means for detecting expiration of the timer;
means for, in response to detecting expiration of the timer, determining if the
module is disabled, and if the module is disabled, enabling the module; and
means for notifying a subscriber in the module of expiration of the timer.

25